REMARKS

Claims 61-63 have been amended. No claims have been added or canceled. Therefore, claims 1-63 are pending in the application.

Objection to the Title:

The Examiner objects to the title as not being descriptive and indicative of the invention to which the claims are directed. Applicants respectfully disagree with the Examiner and assert that the current title, "Secure Access to Managed Network Objects using a Configurable Platform-Independent Gateway", is both descriptive and indicative of Applicants' invention. The Examiner states that the present title is not sufficient for proper classification of the claimed subject matter. Applicants request that the Examiner explain exactly what is lacking from the title that results in an inability to classify the claimed subject matter.

The Examiner has previously suggested adding "CORBA" to the title. However, the present title includes, "a Configurable Platform-Independent Gateway", of which a CORBA gateway is but one example. As the invention is not limited to only CORBA embodiments, adding the word "CORBA" to the title would unreasonably limit the title and thus clearly misrepresent the present invention.

Section 112, Second Paragraph, Rejection:

The Office Action rejected claims 58-63 under 35 U.S.C. § 112, second paragraph, as indefinite. Applicants respectfully traverse the § 112 rejections of claims 58-63 for at least the following reasons.

Claim 58 does not recite the limitation, "the manager access," as asserted by the Examiner. Thus, Applicants respectfully traverse the § 112 rejection of claim 58 and request the removal thereof.

The Examiner rejected claims 59 and 60, arguing that there is insufficient antecedent basis for the phrase, "the manager access." However, claims 59 and 60 both include the phrases "access for the manager application" and "the manager application is granted access", thus making the subsequent use of the phrase, "the manager access" clear and fully understandable to one of skill in the art. Applicants remind the Examiner that, as noted in § 2173.05(e) of the M.P.E.P, "[i]f the scope of a claim would be reasonable ascertainable by those skilled in the art, then the claim is not indefinite." As one skilled in the art would easily ascertain that the phrase, "the manager access" refers to the access granted the manager application, as recited earlier in the claim, Applicants submit that claims 59 and 60 are not indefinite. As such, Applicants respectfully traverse the § 112 rejections of claims 59 and 60 and request the removal thereof.

The Examiner also rejected claims 61-63, arguing that there is insufficient antecedent basis for the phrases, "the insertion of the user name" and "the request message to enforce object-level access control." Claims 61-63 have been amended to include the Examiner's suggested amendments, as will be discussed in more detail below. Applicants assert that the amended claims are in complete compliance with 35 U.S.C. § 112.

Section 102(e) Rejection:

The Office Action rejected claims 1-57 under 35 U.S.C. § 102(e) as being anticipated by Barker et al. (U.S. Patent 6,363,421), hereinafter "Barker."

Regarding claim 1, Applicants respectfully disagree with the Examiner's interpretation of Barker and submit that Barker does not anticipate a gateway that is configurable to provide object-level access control between the managers and the managed objects, wherein said object-level access control is provided at the individual object level so that one of the managers is granted access to one of the managed objects while being prevented from interfacing with a different one of the managed objects.

Instead, Barker discloses a system for "access control based on client name and password" (Barker, column 8, lines 45-46). Barker describes this as "a method of *client based* access control of network elements" (Emphasis added, Barker, column 30, lines 45-46). Further, Barker summarizes his access control features with "the *client based access control* ... provides a means to restrict access on a *command/client basis*", not at the object level. (emphasis added, Barker, column 31, lines 10-12).

The Examiner cites a passage from Barker (column 23, line 55 – column 26, line 10) describing a set of procedures whereby a client may register to receive notification when managed object attribute values change. The Examiner specifically quotes one line that states, "[n]ote that if more than one attribute has changed for a managed object instance, the changes will be grouped and delivered to each registered client on a managed object instance basis" (Barker, column 26, lines 6-10). Although this portion of Barker teaches clients receiving attribute updates from individual managed objects, and hence object-level notification, it does not teach object-level *access control.* Barker explicitly teaches providing client-based access control at the start of a client session (see Barker, column 30, lines 47-52), while not requiring further authentication or access control based upon which managed objects the client wishes to access. The portion of Barker cited by the Examiner has absolutely nothing to do with access *control* provided at the individual object level so that one of the managers is granted access to one of the managed objects while being prevented from interfacing with a different one of the managed objects.

In the Response to Arguments section of the Final Action, the Examiner contends that Barker's use of a managed object identifier for network elements teaches object-level access control and cites Figure 6 of Barker. However, Figure 6 of Barker only teaches that a combination of a managed object class code and an instance identifier defines the managed object identifier. Figure 6 mentions nothing regarding object level access control. Instead, Figure 6 is merely a listing of terms. Barker specifically states, "FIG. 6 is a table of terms associated with the managed object model" (Barker, column 3, lines 12-13). The mere use of the term "managed object identifier" cannot properly be

construed as to disclose, teach, or even suggest controlling access to managed objects via object-level access control as recited in claim 1. Furthermore, the Examiner seems to be implying that if Barker's system includes any mechanism to address individual managed objects, Barker's system necessarily includes object-level access control. This is plainly incorrect. Object-level addressing and object-level access control are two very different things and object-level addressing does not imply object-level access control.

As the Examiner notes, "[e]ach managed object class requires the session identifier as a parameter to each public method" (Barker, column 30, lines 56-58). The session identifier included as a parameter in each public method allows a managed object class to validate the current session – i.e. to ensure that the client has registered with the server and that the session is currently valid. Barker does not teach a client presenting a user name, password or other authentication credentials when registering for object attribute update notification. In other words, Barker does not teach any access control at the object level. Instead, Barker teaches client-level access control where the client must only provide the session ID, object instance identifier, a set of desired attribute codes, and a callback function when registering for attribute update notifications (see Barker, column 25, lines 23-30). In response, the Examiner, in the Response to Arguments section of the Final Action, asserts that, "a client presenting a user name, password or other authentication credentials when registering for object attribute update notification" is not recited any applicants' claims. However, the Examiner has misunderstood applicants' argument. Applicant is arguing that Barker only requires client authentication credentials when initially registering for a session and that Barker does not teach or require a client to present credentials, such as user name, password, etc, when registering for object attribute update notifications, as would be required if Barker actually taught object-level access control. The fact that Barker fails to include any mechanism that could be used to provide access controls on an object-level basis clearly shows that Barker fails to disclose any form of object-level access control. Thus, Barker teaches that a managed object class relies upon the server to perform client authentication by requiring a client to only include a valid session identifier in public method calls without providing or requiring any authentication at the managed object level.

Additionally, Barker teaches that a client can specify a range of managed object instance identifiers, or even *request all instances* in a managed object call through the managed object instance identifier parameter (Barker, column 25, lines 27-28). Hence, Barker teaches that once a client has been properly authenticated at the start of a session, that client may then register for attribute update notification for a number of managed objects through a single call. Such functionality is clearly not compatible with object-level access control, and thus Barker clearly teaches away from object-level access control, wherein the object-level access control is provided at the individual object level so that one of the managers is granted access to one of the managed objects while being prevented from interfacing with a different one of the managed objects.

In response to the Applicants' previous arguments, the Examiner argues that Barker uses "a naming service that provides individual object level access control so that an agent is granted access to an object on the network to support the IIOP protocol" citing column 8, line 53 – column 9, line 19 and column 7, lines 47-63. Applicants note, however, that these passages of Barker only refer to his use of EMAPI, CORBA, Java, C++, and SNMP, but fail to mention anything regarding any sort of access control for any portion of Barker's system. The Examiner has not cited any particular portion in Barker that describes the features the Examiner is attributing to Barker's system. In fact, the Examiner is incorrectly assuming that Barker's use of CORBA and the IIOP protocol includes object level access control such that one of the managers is granted access to one of the managed objects while being prevented from interfacing with a different one of the managed objects.

The Examiner also cites Barker teaching, "access permissions associated with the session are examined before authorizing client execution (e.g. remove operation)" (parenthesis in original) (Barker, column 30, lines 58-60). However, this portion of Barker is clearly referring to ensuring that the client has started a valid session with the server. In fact, Barker, referring to the same remove operation, clearly states, "[a]s with any other client requests, the *client must have created a session prior to performing this*

operation." (Emphasis added, Barker, column 22, lines 51-53). Thus, Barker is clearly referring to client-level access control, not object-level access control.

Barker clearly does not teach object-level access control between the managers and the managed objects. The Examiner contends, in the Response to Arguments section of the Final Action, that Barker's use of a managed object identifier for network elements teaches object-level access control and again cites Figure 6 of Barker. However, Figure 6 of Barker only teaches that a combination of a managed object class code and an instance identifier defines the managed object identifier. As noted above, Figure 6 does not mention anything regarding object level access control. The mere use of an managed object identifier does not disclose, teach, or even suggest controlling access to managed objects via object-level access control. Object identifiers are used in all types of systems, regardless of what type of access control is provided. The Examiner's arguments are completely irrelevant to object-level access control.

For at least the reasons given above, the rejection of claim 1 is not supported by the prior art and its removal is respectfully requested. Similar remarks as those above regarding claim 1 also apply to claims 58 and 61.

Regarding claim 20, the Examiner states, "Barker teaches... wherein the gateway in configured to ... determine on a managed object level whether or not the manager application is allowed to send a request to the managed object as a function of the user of the manager application." Applicants strongly disagree with the Examiner's interpretation of Barker and submit that Barker fails to anticipate determining on a managed object level whether or not the manager application is allowed to send a request to the managed object. In contrast, as shown in the arguments regarding claim 1 above, Barker discloses a method of client-based access control of network elements as a means to restrict access on a command/client basis.

Barker further teaches the use of a single service object "to provide services for a class of managed objects" (Barker, column 14, lines 42-43) and that the EM server "will

implement one application-specific service object for each type of physical or logical resource to be managed" (underlining added) (Barker, column 39, lines 60-62). Applicants assert that access control on a command/client basis while using a single service object for each class of managed object actually teaches away from determining on a managed object level whether or not the manager application is allowed to send a request to the managed object.

For at least the reasons given above, the rejection of claim 20 is not supported by the prior art and its removal is respectfully requested. Similar remarks as discussed above in regard to claim 20 apply to claims 39, 59, 60, 62 and 63.

Regarding claim 2, Barker fails to teach wherein the gateway is configurable to determine whether each of the managers is authorized to communicate with each of the managed objects. Instead, Barker teaches the use of a single service object "to provide services for a class of managed objects" (Barker, column 14, lines 42-43) and that the EM server "will implement one application-specific service object for each type of physical or logical resource to be managed" (underlining added) (Barker, column 39, lines 60-62). Applicants assert that access control on a command/client basis while using a single service object for each class of managed object actually teaches away from determining on a managed object level whether or not the manager application is allowed to send a request to the managed object.

In response, the Examiner cites various pieces of Barker's system in the Response to Arguments section of the Final Action. Specifically, the Examiner refers to Barker's element management server in Figure 1A, software modules of Figures 3 and 4, the term user session of figure 6 and network elements of figure 1C. However, the Examiner completely failed to cite any portion of Barker that actually teaches or suggests that the various, disparate, elements of Barker's system cited by the Examiner actually perform or teach a gateway configurable to determine whether each of the managers is authorized to communicate with each of the managed objects. The fact that Barker includes various system elements that may correspond to certain elements recited in claim 2 does not

imply that they are arranged as in claim 2 nor that perform specific limitations as recited in claim 2.

Additionally, Barker discloses client-based access control that provides a means to restrict access on a command/client basis (Barker, column 31, lines 10-12). Hence, Barker teaches access control based on a command/client basis, not a managed object basis and thus fails to disclose a gateway that is configurable to determine whether each of the managers is authorized to communicate with each of the managed objects.

For at least the reasons given above, the rejection of claim 2 is not supported by the prior art and its removal is respectfully requested. Similar remarks as discussed above in regard to claim 2 apply to claims 21, and 40.

Regarding claim 3, Barker fails to teach a gateway that is configurable to authenticate the managers to receive the events from or to send the request to the managed objects as a function of the identity of the managed object as the Examiner asserts. As the Examiner states, Barker teaches the use of basic server authentication, SSL, and web server administration including client name and password for access control (Barker, column 8, lines 31-54). Further, Barker discloses client based access control that provides a means to restrict access on a command/client basis (Barker, column 31, lines 10-12). However, basic server authentication and SSL using client names and passwords do not imply authenticating managers as a function of the identity of the managed object.

In response, the Examiner cites various pieces of Barker's system in the Response to Arguments section of the Final Action. Specifically, the Examiner refers to Barker's element management server in Figure 1A, software modules of Figures 3 and 4, and the term "notification" in figure 6. However, the Examiner completely failed to cite any portion of Barker that actually teaches or suggests that the various, disparate, elements of Barker's system cited by the Examiner actually include or teach a gateway configurable to authenticate the managers to receive the events from or to send the request to the

managed objects as a function of the identity of the managed object. The fact that Barker includes various system elements that may correspond to certain elements recited in claim 3 does not imply that they are arranged as in claim 3 nor that perform specific limitations as recited in claim 3.

For at least the reasons given above, the rejection of claim 3 is not supported by the prior art and its removal is respectfully requested. Similar remarks as discussed above in regard to claim 3 apply to claims 22, and 41.

In regard to claim 8, Barker fails to teach wherein the managed objects comprise one or more objects *corresponding* to a telephone network, as asserted by the Examiner. In contrast, Barker discloses a system client that is connected to a network element and element management system client *through* a public switched telephone network (Barker, column 3, lines 48-53). Additionally, Barker teaches the use of a telephone system network through the computer internet and a telephonic link for a system client to connect to the system server (Barker, column 3, lines 54-62). The Examiner argues that Barker's use of the phrase "network elements of a telecommunication network" (See, Barker, Title, and brief descriptions of FIG 1A, 1B, and 1C, column 2, lines 50-65) imply that one or more of Barker's network elements correspond to a telephone network. Applicants submit, however, that Barker is referring to network element residing *on* a telecommunications network, not an object *corresponding to* a telephone network. For instance, when discussing FIG. 1B, Barker describes his system as a "method for managing the network element 14 *in* a telephonic network" and continues, "[n]etwork element 14 is connected *through* a telephonic computer network 35 to a computer internet 36" (emphasis added, Barker, column 3, lines 53-58).

In response, the Examiner, in the Response to Arguments section, cites network element 14 of Figure 1A. However, as noted above, network element 14 is not illustrated as corresponding to a telephone network, but rather network element 14 is illustrated as coupled to and communicating over a telephone network. Hence, Barker discloses *using* a telephonic connection between clients and servers but fails to disclose anything

regarding managed objects comprising one or more objects *corresponding to* a telephone network. This is made clear when figures 1A, 1B and 1C are viewed together. It is very clear that Barker is illustrating that fact that his system may be implemented (e.g. his element management server may communicate with network elements) over various types of communication networks, such as public switched telephone network 33 (Figure 1A), telephonic system network 37 (Figure 1B), and a local area network (Figure 1C).

None of the managed objects in Barker *correspond to a telephone network* themselves, but instead communicate using a telephone network. Thus, Barker does not teach wherein the managed objects comprise one or more objects corresponding to a telephone network.

For at least the reasons given above, the rejection of claim 8 is not supported by the prior art and its removal is respectfully requested. Similar remarks as discussed above in regard to claim 8 apply to claims 27, and 46.

Regarding claim 10, the Examiner contends that Barker teaches a gateway that is configurable to provide security audit trails. Applicants disagree with the Examiner and submit that at the Examiner's cited passage (Barker, column 17, line 27 – column 18, line 67) Barker only refers to auditing when describing the clean up of filters for a removed client session. For instance, Barker states, "when the Client Session Manger removes a session and/or application from its internal structures, it notifies the Event Distributor via a callback, at which point the Event Distributor removes all filters associated with the session and/or application" (Barker, column 18, lines 10-18). Thus, Barker refers to active Auditing of client sessions to facilitate clean up of event filter lists, but does not include anything about providing security audit trails.

Therefore, the rejection of claim 10 is not supported by the prior art and its removal is respectfully requested. Similar remarks as discussed above in regard to claim 10 apply to claims 29 and 48.

In regard to claim 11, Barker fails to teach that the gateway providing security audit trails comprises the gateway providing access to a logging service. As shown in the arguments above regarding claim 10, Barker fails to teach a gateway providing security audit trails. Barker also fails to teach that the gateway providing security audit trails comprises the gateway providing access to a logging service. The Examiner cites passages referring to individual components of Barker's system storing lists of events to storage devices (Barker, column 11, lines 18-60, column 17, line 33-column 18, line 9, and column 41, line 63 – column 42, line 53). However, Applicants submit that individual components using storage devices to maintain their own lists of data does not equate to a gateway providing access to a logging service.

In the Response to Arguments section, the Examiner merely cites the same passages of Barker as cited in the rejection of claim 11 without providing any additional explanation or argument regarding his interpretation regarding the teaches of the cited passages. Applicants maintain that the cited passages do teach a gateway providing access to a logging service. As noted above, individual components storing event lists to their own storage devices clearly fails to teach a gateway providing access to a logging service.

For at least the reasons given above, the rejection of claim 11 is not supported by the prior art and its removal is respectfully requested. Similar remarks as discussed above in regard to claim 11 apply to claims 30, and 49.

Regarding claim 12, the Examiner contends that Barker teaches, "the logging service (local data services at the server) is operable to log an ID of a user that sends each request" (parenthesis and underlining in original). Applicants respectfully disagree with the Examiner's interpretation of Barker. The Examiner cites the same passage cited in regard to the rejection of claim 11 above, but Applicants note that these passages merely refer to the fact that Barker's system includes the ID of a client application when registering event filters for that client application. However, such use of the client application ID does not imply that Barker provides access to a logging service operable to

log an ID of a user that receives each event or sends each request. Instead, Barker teaches that his Event Distributor provides an IDL interface for registering filters based in part on an application ID (Barker, column 17, lines 28-30).

In response to the above argument, the Examiner, in the Response to Arguments section of the Final Action, argues that "it is noted that the features upon which applicant relies, 'the logging service, local data services at the server, is operable to log an ID of a user that sends each request' are not recited in the rejected claim(s)." The Examiner has either misread or misunderstood Applicants' argument. Applicants are not arguing that the specific phrase is recited in any claim, but instead Applicants are refuting the Examiner's assertion regarding Barker's teachings. Specifically, as argued previously and above, the Examiner contends that Barker teaches a logging service, which the examiner is presumably equating to Barker's local data services at the server, that is operable to log an ID of a user that sends each request. Applicants' disagree with the Examiner, as noted above. Thus, Examiner's Response to Argument regarding the fact that "local data services at the server" is not recited in Applicants' claims is completely irrelevant to Applicants' arguments.

Thus, for at least the reasons given above, the rejection of claim 12 is not supported by the prior art and its removal is respectfully requested. Similar remarks as discussed above in regard to claim 12 apply to claims 31, and 50.

Regarding claim 18, Barker fails to anticipate wherein requests are converted from the interface definition language to a Portable Management Interface (PMI) format prior to delivery to the managed objects. Instead, Barker teaches, "SNMP Mediator 160 provides translation between the MIB ASN.1 format and the managed object notation used in this architecture" (Barker, column 11, lines 39-42). The Examiner cites a passage (column 21, line 46 - column 22 line 59) where Barker notes that new managed objects could be added (to his system) that utilize a different protocol and encapsulate that knowledge in the managed object class (Barker, column 22, lines18-20). However, Barker fails to mention the PMI format. The Examiner is apparently arguing that by

simply stating that other formats may be used, Barker is specifically anticipating every other possible format, including PMI. This is clearly an incorrect interpretation of Barker's teachings. Therefore, Applicants submit that Barker fails to teach that the requests are converted from the interface definition language to a Portable Management Interface (PMI) format prior to delivery to the managed objects as contended by the Examiner.

In the Response to Arguments section, the Examiner merely repeats the rejection and again cites column 21, line 46 to column 22 line 59 of Barker. Thus, the Examiner fails to provide any additional argument or explanation regarding his contention that Barker specifically anticipates converting requests from the interface definition language to PMI. Without specific teachings by Barker regarding converting requests into PMI, Barker clearly fails to *anticipate* such conversions. As the Examiner should be aware, "[a] claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference" (See, M.P.E.P. 2131). Barker clearly fails to expressly or inherently describe converting requests to PMI.

Thus, the rejection of claim 18 is not supported by the prior art and its removal is respectfully requested. Similar remarks as discussed above in regard to claim 18 apply to claims 37, and 56.

Regarding claim 19, the Examiner states, "Barker teaches ... the requests are converted from the interface definition language to a platform-specific format prior to delivery to the managed objects," citing Barker's SNMP mediator providing translation between the MIB ASN.1 format and the managed object notation used in this architecture. Applicants respectfully disagree with the Examiner's interpretation of Barker.

Barker teaches the use of SNMP as the communication protocol between element management system and the managed elements (Barker, column 4, lines 43-45).

Applicants assert the SNMP is not a platform-specific format, but rather is a network protocol that contains no platform specific features. Thus, Barker fails to teach the requests are converted from the interface definition language to a platform-specific format prior to delivery to the managed objects as asserted by the Examiner.

For at least the reasons given above, the rejection of claim 19 is not supported by the prior art and its removal is respectfully requested. Similar remarks as discussed above in regard to claim 19 apply to claims 38, and 57.

The Examiner rejected claims 58-63 under 35 U.S.C. § 102(e) as being anticipated by Vuong et al. (U.S. Patent 6,430,578), hereinafter "Vuong." Applicants respectfully traverse this rejection in light of the following remarks.

Regarding claim 58, contrary to the Examiner's assertion, Vuong fails to disclose a gateway which is coupled to a plurality of managed objects and which is configured to deliver events generated by the managed objects to one or more managers or to deliver requests generated by the managers to one or more of the managed objects. Vuong teaches a naming service that provides unique identifiers and addresses for processes on a computer network. Vuong's name service includes a database of the identifiers and addresses and the name service response to queries by searching the database and returning any results. (Vuong, Abstract; column 2, lines 7-15). The Examiner cites column 5, line 57 – column 6, line 23. However, the cited passage describes how Vuong's name service accepts names from agents on the computer network and, after determining whether or not the name is unique, either adds the agent's name to the name service's database or sends a "refuse request" message to the agent. The cited passage does not mention any gateway coupled to a plurality of managed objects. Database entries are not managed objects, as managed objects are understood in the art. Presumably the Examiner interprets Vuong's name service as a gateway. However, Vuong's name service is not coupled to a plurality of managed objects. Instead, Vuong's name service merely responds to requests to add to and queries to

retrieve information from the name service's database. Even if one could interpret Vuong's name service as a managed object, which Applicants maintain one cannot, the database is clearly not managed by the requesting agents. Merely requesting that a name and/or address be inserted as an entry into the database does not constitute managing the database. Clearly Vuong's name service manages the database. In fact, Vuong very clearly states, "Name Service 112 *maintains* a database holding identification and addressing information" and "the database *controlled by* the Name service is an object-oriented database" (emphasis added, Vuong, column 3, lines 57-63). Thus, Vuong teaches that his name service controls and maintains the database.

Additionally, the agents registering their names with Vuong's name service are not managers and do not generate requests to managed objects. Instead, Vuong's agents merely request that their name (and address) be included in the name service's database. Vuong does not teach that an agent registering its name with the name service is a manager generating requests to a managed object. Instead, as noted above, Vuong's name service maintains and controls the database.

Vuong also fails to disclose a gateway configurable to provide object-level access control between the managers and the managed objects. The Examiner cites column 2, lines 26-52 and column 6, lines 42-59 of Vuong. The first cited passage provides an introduction to Vuong's name service for "managing names and identities of processes running on a computer network" (Vuong, column 2, lines 26-28). This passage further describes how Vuong's name service includes a receiver that accepts a name from a process on the computer network and a comparator configured to determine whether the process is a component of the computer management infrastructure for the computer network. The second cited passage (Vuong, column 6, lines 42-59) describes the ability of Vuong's name service to respond to "relatively sophisticated queries." For example, Vuong's query syntax supports prefixes, suffixes, infixed, and full or partial names using wildcards. This passage further describes how registered entities may receive updates or changes made to the name service's database.

Nowhere in either cited passage, nor in fact in the entire Vuong reference, is there any mention of a gateway configured to provide object-level access control between managers and managed objects. Instead, Vuong provides a name service that collects, maintains, and disseminates unique identifiers and addresses for processes on a computer network. Providing identifiers and addresses for processes on a computer network is clearly not the same as providing object-level access control between managers and managed objects. Vuong does not mention any sort of access control in his name service. The Examiner seems to be implying that any form of object-level access necessarily includes object-level access *control* at the individual object level so that one of the managers is granted access to one of the managed objects while being prevented from interfacing with a different one of the managed objects. However, object-level access can be provided with or without imposing *access control.* In Vuong, no form of access control is disclosed or contemplated.

Furthermore, Vuong fails to disclose wherein the object-level access control is provided at the individual object level so that one of the managers is granted access to one of the managed objects while being prevented from interfacing with a different one of the managed objects. The Examiner again cites column 2, lines 26-52 and column 6, lines 42-59 of Vuong. However, neither of these passages mentions anything regarding a agent, which the Examiner is presumably interpreting as a manager, being granted access to one database entry, which the Examiner is presumably interpreting as a managed object, while being prevented from interfacing with a different one of the database entries. Instead, the cited passages describe how Vuong's name service responds to queries. Vuong doesn't mention anything regarding preventing access to his database on an entry-level basis.

Applicants submit that the rejection of claim 58 is clearly not supported by the cited art and removal thereof is respectfully requested.

Regarding claim 59, Vuong fails to disclose determining on a managed object level whether or not the manager application is allowed to receive an event generated by

one of a plurality of managed objects or to send a request to the one of the plurality of managed objects as a function of the identity of the user of the manager application. The Examiner cites column 7, lines 9-32. However, the cited reference has absolutely no relevance to determining, as a function of the identity of a user of the manager application whether or not the manager application is allowed to receive an event generated by or to send a request to one of a plurality of managed object. Instead, the cited reference merely describes how an agent, or other entity on the computer network, can de-register with Vuong's name service and thereby remove its name from the name service's database. The cited reference makes not mention to determining whether or the requesting agent can access a managed object. Even if one interprets the entries of Vuong's database as managed object, which Applicants maintain one cannot, the cited passage still does not disclose anything regarding determining whether or not the de-registering agent can access the database entry. Instead, Vuong teaches only that the name service checks the agent's name against the database and if it is found, the entry is removed.

Vuong also fails to disclose whereby access for the manager application to receive the event or send the request is approved or denied for said one of the plurality of managed objects at the individual object level so that the manager application is granted access to one of the plurality of managed objects while being prevented from interfacing with a different one of the plurality of managed objects, contrary to the Examiner contention. The Examiner cites column 8, lines 21-42 of Vuong. Applicants can see no relevance of the cited passage. The cited passage discusses the "various devices and entities" that reside on and communicate over a computer network. Vuong mentions devices and entities such as client computers, data storage devices, modems, printers, hubs, routers, packet switches, hosts, and bridges. The cited passage is, however, completely silent regarding approving or denying access for a manager application at an individual object level so that the manager application is granted access to one while being prevented from interfacing with a different one of a plurality of managed objects. The Examiner seems to be arguing that merely listing various devices that may reside and communicate on a computer network implies providing such access control at an

individual object level. The Examiner is clearly inserting his own assumptions and speculation into Vuong's system in hindsight.

Thus, the rejection of claim 59 is not supported by the cited prior art and removal thereof is respectfully requested. Similar remarks as those above regarding claim 59 also apply to claim 60.

Regarding claim 61, contrary to the Examiner's assertion, Vuong fails to disclose a gateway which is coupled between a plurality of managed objects and a plurality of proxy agent managers; and which is configured to deliver events generated by the managed objects to one or more managers or to deliver requests generated by the managers to one or more of the managed objects. Vuong teaches naming service that provides unique identifiers and addresses for processes on a computer network. Vuong's name service includes a database of the identifiers and addresses and the name service response to queries by searching the database and returning any results. (Vuong, Abstract; column 2, lines 7-15). The Examiner cites column 5, line 57 – column 6, line 23. However, the cited passage describes how Vuong's name service accepts names from agents on the computer network and, after determining whether or not the name is unique, either adds the agent's name to the name service's database or sends a "refuse request" message to the agent. The cited passage does not mention any gateway coupled to a plurality of managed objects. Database entries are not managed objects, as managed objects are understood in the art. Presumably the Examiner interprets Vuong's name service as a gateway. However, Vuong's name service is not coupled between a plurality of managed objects and a plurality of proxy agent managers. Instead, Vuong's name service merely responds to requests to add to and queries to retrieve information from the name service's database. Even if one could interpret Vuong's name service as a managed object, which applicants maintain one cannot, the database is clearly not managed by the requesting agents. Merely requesting that a name and/or address be inserted as an entry into the database does not constitute managing the database. Clearly Vuong's name service managed the database. In fact, Vuong very clearly states, "Name Service 112 *maintains* a database holding identification and addressing information" and "the

database *controlled by* the Name service is an object-oriented database" (emphasis added, Vuong, column 3, lines 57-63). Thus, Vuong teaches that his name service controls and maintains the database.

Vuong further fails to disclose wherein each of the events and each of the requests include a user identification, wherein the user identification identifies the respective manager to which the event or the request belongs.

Vuong also fails to disclose a gateway configurable to provide object-level access control between the managers and the managed objects. The Examiner cites column 2, lines 26-52 and column 6, lines 42-59 of Vuong. The first cited passage provides an introduction to Vuong's name service for "managing names and identities of processes running on a computer network" (Vuong, column 2, lines 26-28). This passage further describes how Vuong's name service includes a receiver that accepts a name from a process on the computer network and a comparator configured to determine whether the process is a component of the computer management infrastructure for the computer network. The second cited passage (Vuong, column 6, lines 42-59) describes the ability of Vuong's name service to respond to "relatively sophisticated queries." For example, Vuong's query syntax supports prefixes, suffixes, infixed, and full or partial names using wildcards. This passage further describes how registered entities may receive updates or changes made to the name service's database.

Nowhere in either cited passage, nor in fact in the entire Vuong reference, is there any mention of a gateway configured to provide object-level access control between managers and managed objects. Instead, Vuong provides a name service that collects, maintains, and disseminates unique identifiers and addresses for processes on a computer network. Providing identifiers and addresses for processes on a computer network is clearly not the same as providing object-level access control between managers and managed objects. Vuong does not mention any sort of access control in his name service. The Examiner seems to be implying that any form of object-level access necessarily includes object-level access *control*. However, object-level access can be provided with

or without imposing *access control*. In Vuong, no form of access control is disclosed or contemplated.

Vuong also fails to mention anything regarding wherein the managers share a singleton Request Service Access Point (Request SAP) object.

Furthermore, Vuong fails to disclose wherein the object-level access control is provided by the Request SAP object at the individual object level so that one of the managers is granted access to one of the managed objects while being prevented from interfacing with a different one of the managed objects. The Examiner again cites column 2, lines 26-52 and column 6, lines 42-59 of Vuong. However, neither of these passages mentions anything regarding a agent, which the Examiner is presumably interpreting as a manager, being granted access to one database entry, which the Examiner is presumably interpreting as a managed object, while being prevented from interfacing with a different one of the database entries. Instead, the cited passages describe how Vuong's name service responds to queries. Vuong doesn't mention anything regarding preventing access to his database on an entry-level basis.

Thus, the rejection of claim 59 is not supported by the cited prior art and removal thereof is respectfully requested.

Regarding claim 62, Vuong fails to disclose determining on a managed object level whether or not the manager application is allowed to receive an event generated by one of a plurality of managed objects or to send a request to the one of the plurality of managed objects as a function of the identity of the user of the manager application. The Examiner cites column 7, lines 9-32. However, the cited reference has absolutely no relevance to determining, as a function of the identity of a user of the manager application whether or not the manager application is allowed to receive an event generated by or to send a request to one of a plurality of managed object. Instead, the cited reference merely describes how an agent, or other entity on the computer network, can de-register with Vuong's name service and thereby remove its name from the name

service's database. The cited reference makes not mention to determining whether or the requesting agent can access a managed object. Even if one interprets the entries of Vuong's database as managed object, which applicants maintain one cannot, the cited passage still does not disclose anything regarding determining whether or not the de-registering agent can access the database entry. Instead, Vuong teaches only that the name service checks the agent's name against the database and if it is found, the entry is removed.

Vuong also fails to disclose whereby access for the manager application to receive the event or send the request is approved or denied for said one of the plurality of managed objects by a singleton Request Service Access Point (Request SAP) at the individual object level so that the manager application is granted access to one of the plurality of managed objects while being prevented from interfacing with a different one of the plurality of managed objects, contrary to the Examiner contention. The Examiner cites column 8, lines 21-42 of Vuong. Applicants can see no relevance of the cited passage. The cited passage discusses the "various devices and entities" that reside on and communicate over a computer network. Vuong mentions devices and entities such as client computers, data storage devices, modems, printers, hubs, routers, packet switches, hosts, and bridges. The cited passage is, however, completely silent regarding approving or denying access for a manager application at an individual object level so that the manager application is granted access to one while being prevented from interfacing with a different one of a plurality of managed objects. The Examiner seems to be arguing that merely listed various devices that may reside and communicate on a computer network implies providing such access control at an individual object level. The Examiner is clearly inserting his own assumptions and speculation into Vuong's system in hindsight.

Furthermore, Vuong fails to disclose wherein the event and the request include a user identification, wherein the user identification identifies the manager application to which the event or the request belongs. Nowhere does Vuong mention including user identification in each event and request. Vuong also fails to mention anything regarding object-level access control provided by a Request SAP object. The Examiner does not

cite any portion of Vuong that teaches that object-level access control is provided by a Request SAP object.

Thus, the rejection of claim 62 is not supported by the cited prior art and removal thereof is respectfully requested. Similar remarks as those above regarding claim 62 also apply to claim 63.

The Office Action rejected claims 58-63 under 35 U.S.C. § 102(e) as being anticipated by Spencer (U.S. Patent 6,253,243). Applicants respectfully traverse this rejection in light of the following remarks.

Regarding claim 58, Spencer fails to disclose a gateway that is configurable to provide object-level access control between the managers and the managed objects to receive the events from or to send the requests to the managed objects, contrary to the Examiner's assertion. The Examiner cites a passage (column 5, lines 46-65) where Spencer describes how a user-developed management application 300 communicates with MIS server 306 via a portable management interface (PMI) 302. Spencer describes how PMI 302 is an object-oriented interface that provides access to management information. The cited passage does not teach anything about a gateway providing object-level access control between managers and managed objects. The Examiner has not provided any argument or explanation regarding his interpretation of the cited passage.

Spencer further fails to disclose wherein the object-level access control is provided at the individual object level so that one of the managers is granted access to one of the managed objects while being prevented from interfacing with a different one of the managed objects, contrary to the Examiner's contention. The Examiner cites column 7, lines 35-57 of Spencer. However, the cited passage teaches how Spencer's SNMP trap system extracts the IP address from an <agent_addr> field of the SNMP trap Protocol Data Unit (PDU). The PDU is the format for SNMP trap data in Spencer's

system. After extracting the IP address, Spencer's system determines if there is an object configured to represent that agent system. If such an object is found, the trap's originating system's cmipsnmpProxyAgent instance is set as the source object instance for the trap alarm. Thus, the cited passage not only fails to mention anything about object-level access control, it has no relevance to object-level access control. The cited passage does not teach anything about providing object-level access control at the individual object level so that a manager is granted access to one managed object while being prevented from interfacing with a different one of the managed objects.

Thus, for at least the reasons presented above, the rejection of claim 58 is not supported by the prior art and removal thereof is respectfully requested.

Regarding claim 59, contrary to the Examiner's assertion, Spencer fails to disclose sending an identity of a user of a manager application to a gateway. The Examiner cites column 7, lines 35-67 of Spencer. However the cited passage makes no mention of sending an identity of a *user of a manger* application to a gateway. Instead, the cited passage describes how Spencer's system uses an IP address to locate a proxy agent object to represent a SNMP trap's agent system. Nowhere does Spencer mention sending an identity of a user of a manager application to a gateway.

Additionally, Spencer fails to disclose determining on a managed object level whether or not the manager application is allowed to receive an event generated by one of a plurality of managed objects or to send a request to the one of the plurality of managed objects as a function of the identity of the user of the manager application, contrary to the Examiner's contention. The Examiner cites column 5, line 53-column 6, line 13. However, the cited passage does not teach or even mention determining on a managed object level whether or note the manager application is allowed to receive an event generated by or to send a request to one of the plurality of managed objects as a function of the identity of the user of the manager application. Instead, the cited passage describes how a managed application 300 communicates with an MIS server according to the

portable management interface and how portable management interface is able to access managed object instance state information, class schema, and event services. Spencer does not discuss or mention anything about an identity for a user of a manager application. Nor does Spencer mention determining whether or not the manager application can send requests or send events to managed objects as a function of the identity of the user of the manager application.

Spencer also fails to disclose whereby access for the manager application to receive the event or send the request is approved or denied for said one of the plurality of managed objects at the individual object level so that the manager application is granted access to one of the plurality of managed objects while being prevented from interfacing with a different one of the plurality of managed objects, contrary to the Examiner assertion. The Examiner again cites column 7, lines 35-67 of Spencer. However, as noted above, this passage does not mention any sort of object-level access control. The cited passage further fails to mention anything regarding approving or denying the manager application access to receive an event or send a request at the individual object level. The cited passage fails to mention any sort of access control whatsoever. The Examiner has clearly misunderstood or misinterpreted the teachings of Spencer.

For at least the reasons presented above, the rejection of claim 59 is not supported by the prior art and removal thereof is respectfully requested. Similar remark as those above regarding claim 59 also apply to claim 60.

Regarding claim 61, Spencer fails to disclose a gateway that is configurable to provide object-level access control between the managers and the managed objects to receive the events from or to send the requests to the managed objects, contrary to the Examiner's assertion. The Examiner cites a passage (column 5, lines 46-65) where Spencer describes how a user-developed management application 300 communicates with MIS server 306 via a portable management interface (PMI) 302. Spencer describes how PMI 302 is an object-oriented interface that provides access to management information. The cited passage does not teach anything about a gateway providing

object-level access control between managers and managed objects. The Examiner has not provided any argument or explanation regarding his interpretation of the cited passage.

Furthermore, Spencer fails to disclose wherein the event and the request include a user identification, wherein the user identification identifies the manager application to which the event or the request belongs. Nowhere does Spencer mention including user identification in each event and request.

Spencer also fails to mention anything regarding wherein the managers share a singleton Request Service Access Point (Request SAP) object.

Spencer further fails to disclose wherein the object-level access control is provided by the Request SAP object at the individual object level so that one of the managers is granted access to one of the managed objects while being prevented from interfacing with a different one of the managed objects, contrary to the Examiner's contention. The Examiner cites column 7, lines 35-57 of Spencer. However, the cited passage teaches how Spencer's SNMP trap system extracts the IP address from an <agent_addr> field of the SNMP trap Protocol Data Unit (PDU). The PDU is the format for SNMP trap data in Spencer's system. After extracting the IP address, Spencer's system determines if there is an object configured to represent that agent system. If such an object is found, the trap's originating system's cmipsnmpProxyAgent instance is set as the source object instance for the trap alarm. Thus, the cited passage not only fails to mention anything about object-level access control, it has no relevance to object-level access control. The cited passage does not teach anything about providing object-level access control at the individual object level so that a manager is granted access to one managed object while being prevented from interfacing with a different one of the managed objects.

Thus, for at least the reasons presented above, the rejection of claim 61 is not supported by the prior art and removal thereof is respectfully requested.

Regarding claim 62, contrary to the Examiner's assertion, Spencer fails to disclose sending an identity of a user of a manager application to a gateway. The Examiner cites column 7, lines 35-67 of Spencer. However the cited passage makes no mention of sending an identity of a *user of a manger* application to a gateway. Instead, the cited passage describes how Spencer's system uses an IP address to locate a proxy agent object to represent a SNMP trap's agent system. Nowhere does Spencer mention sending an identity of a user of a manager application to a gateway.

Additionally, Spencer fails to disclose determining on a managed object level whether or not the manager application is allowed to receive an event generated by one of a plurality of managed objects or to send a request to the one of the plurality of managed objects as a function of the identity of the user of the manager application, contrary to the Examiner's contention. The Examiner cites column 5, line 53-column 6, line 13. However, the cited passage does not teach or even mention determining on a managed object level whether or note the manager application is allowed to receive an event generated by or to send a request to one of the plurality of managed objects as a function of the identity of the user of the manager application. Instead, the cited passage describes how a managed application 300 communicates with an MIS server according to the portable management interface and how portable management interface is able to access managed object instance state information, class schema, and event services. Spencer does not discuss or mention anything about an identity for a user of a manager application. Nor does Spencer mention determining whether or not the manager application can send requests or send events to managed objects as a function of the identity of the user of the manager application.

Spencer also fails to disclose whereby access for the manager application to receive the event or send the request is approved or denied for said one of the plurality of managed objects at the individual object level so that the manager application is granted access to one of the plurality of managed objects while being prevented from interfacing with a different one of the plurality of managed objects, contrary to the Examiner

assertion. The Examiner again cites column 7, lines 35-67 of Spencer. However, as noted above, this passage does not mention any sort of object-level access control. The cited passage further fails to mention anything regarding approving or denying the manager application access to receive an event or send a request at the individual object level. The cited passage fails to mention any sort of access control whatsoever.

For at least the reasons presented above, the rejection of claim 62 is not supported by the prior art and removal thereof is respectfully requested. Similar remark as those above regarding claim 62 also apply to claim 63.

The Office Action rejected claims 58-63 under 35 U.S.C. § 102(e) as being anticipated by Barker. Applicants respectfully traverse this rejection in light of the following remarks.

Regarding claim 58, Barker does not anticipate a gateway that is configurable to provide object-level access control between the managers and the managed objects, wherein said object-level access control is provided at the individual object level so that one of the managers is granted access to one of the managed objects while being prevented from interfacing with a different one of the managed objects. Instead, as noted above, Barker discloses a system for "access control based on client name and password" (Barker, column 8, lines 45-46). Barker describes this as "a method of *client based* access control of network elements" (Emphasis added, Barker, column 30, lines 45-46). Further, Barker summarizes his access control features with "the *client based access control* ... provides a means to restrict access on a *command/client basis*", not at the object level. (emphasis added, Barker, column 31, lines 10-12). Please refer to the remarks above regarding claim 1 for a more detailed discussion regarding this argument. Furthermore, since claim 58 recites similar limitations as those recited in claim 1, the arguments presented above regarding the rejection of claim 1 in view of Barker also apply to claim 58 with equal force.

Thus, Applicants assert that Barker does not teach object-level access control between the managers and the managed objects. For at least the reasons given above, the rejection of claim 58 is not supported by the prior art and its removal is respectfully requested.

Regarding claim 59, the Examiner states, "Barker teaches... wherein the gateway in configured to ... determine on a managed object level whether or not the manager application is allowed to send a request to the managed object as a function of the user of the manager application." Applicants disagree with the Examiner's interpretation of Barker and submit that Barker fails to anticipate determining on a managed object level whether or not the manager application is allowed to send a request to the managed object. In contrast, as shown in the arguments regarding claim 1 above, Barker discloses a method of client based access control of network elements as a means to restrict access on a command/client basis. For a more detailed discussion regarding this argument, please refer to the remarks above regarding the rejection of claim 20. Furthermore, as claim 59 recites limitations similar to those recited in claim 20, the arguments presented above regarding claim 20 apply to claim 59 with equal force.

For at least the reasons given above, the rejection of claim 59 is not supported by the prior art and its removal is respectfully requested. Similar remarks as discussed above in regard to claim 59 apply to claim 60.

Regarding claim 61, Barker does not anticipate a gateway that is configurable to provide object-level access control between the managers and the managed objects, wherein said object-level access control is provided by the Request SAP object at the individual object level so that one of the managers is granted access to one of the managed objects while being prevented from interfacing with a different one of the managed objects. Instead, as noted above regarding claims 1 and 58, Barker discloses a system for "access control based on client name and password" (Barker, column 8, lines 45-46). Barker describes this as "a method of *client based* access control of network

elements" (Emphasis added, Barker, column 30, lines 45-46). Further, Barker summarizes his access control features with "the *client based access control* ... provides a means to restrict access on a *command/client basis*", not at the object level. (emphasis added, Barker, column 31, lines 10-12). Please refer to the remarks above regarding claim 1 for a more detailed discussion regarding this argument.

Additionally, Barker fails to disclose wherein each of the events and each of the requests include a user identification, wherein the user identification identifies the respective manager to which the event or the request belongs. Nowhere does Barker mention including user identification in each event and request. Barker also fails to mention anything regarding object-level access control provided by a Request SAP object. The Examiner does not cite any portion of Barker that teaches that object-level access control is provided by a Request SAP object.

Barker further fails to disclose wherein the managers share a singleton Request Service Access Point (Request SAP) object. Nowhere does Barker mention, nor has the Examiner cited any passage that mentions, anything regarding a singleton Request SAP object shared by the managers.

Thus, for at least the reasons given above, the rejection of claim 61 is not supported by the prior art and its removal is respectfully requested.

Regarding claim 62, the Examiner states, "Barker teaches... wherein the gateway in configured to ... determine on a managed object level whether or not the manager application is allowed to send a request to the managed object as a function of the user of the manager application." Applicants disagree with the Examiner's interpretation of Barker and submit that Barker fails to anticipate determining on a managed object level whether or not the manager application is allowed to send a request to the managed object. In contrast, as shown in the arguments regarding claim 1 above, Barker discloses a method of client based access control of network elements as a means to restrict access on a command/client basis. For a more detailed discussion regarding

this argument, please refer to the remarks above regarding the rejection of claim 20. Furthermore, as claim 62 recites limitations similar to those recited in claim 20, the arguments presented above regarding claim 20 apply to claim 62 with equal force.

Additionally, Barker fails to disclose wherein the event and the request include a user identification, wherein the user identification identifies the manager application to which the event or the request belongs. Nowhere does Barker mention including user identification in each event and request. Barker also fails to mention anything regarding object-level access control provided by a Request SAP object. The Examiner does not cite any portion of Barker that teaches that object-level access control is provided by a Request SAP object.

For at least the reasons given above, the rejection of claim 62 is not supported by the prior art and its removal is respectfully requested. Similar remarks as discussed above in regard to claim 62 apply to claim 63.

Applicant also asserts that numerous ones of the dependent claims recite further distinctions over the cited art. However, since the rejections have been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time.

Examiner's Proposed Amendments

The Examiner, in both the Final Action and a facsimile communication dated January 20, 2005, has suggested certain amendments to speed up prosecution of the present case. Correspondingly, Applicants have amended claims 61-63 as suggested by the Examiner on pages 19 and 20 of the Final Action. Since the amendments to claims 61-63 are based on the Examiner's recommendations, Applicants submit that entry of these amendments 63 should not raise any new issues. Applicants assert that claims 61-63 are in condition for allowance.

CONCLUSION

Applicants submit the application is in condition for allowance, and notice to that effect is respectfully requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-48400/RCK.

Also enclosed herewith are the following items:

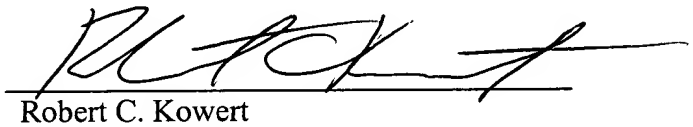
☒ Return Receipt Postcard

☐ Petition for Extension of Time

☐ Notice of Change of Address

☐ Other:

Respectfully submitted,



Robert C. Kowert
Reg. No. 39,255
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: April 6, 2005